

Assignment #D: 十全十美

Updated 1254 GMT+8 Dec 17, 2024

2024 fall, Compiled by 昂奕,化学与分子工程学院

说明:

- 1) 请把每个题目解题思路 (可选), 源码Python, 或者C++ (已经在Codeforces/Openjudge上AC), 截图 (包含Accepted), 填写到下面作业模版中 (推荐使用 typora <https://typoraio.cn>, 或者用word)。AC 或者没有AC, 都请标上每个题目大致花费时间。
- 2) 提交时候先提交pdf文件, 再把md或者doc文件上传到右侧“作业评论”。Canvas需要有同学清晰头像、提交文件有pdf、“作业评论”区有上传的md或者doc附件。
- 3) 如果不能在截止前提交作业, 请写明原因。

1. 题目

02692: 假币问题

brute force, <http://cs101.openjudge.cn/practice/02692>

思路:

初始时所有硬币标记为0,表示未知

先处理even的称量,如果even了就天平两端的所有硬币都标记为'0'(区别于0),表示重量正常;

然后处理up和down,轻的那一边所有硬币-1,重的那边+1,没称的肯定都正常,标'0'.

三次下来,必然只有1个不为0的最值,这个就是假币.

代码:

```
from collections import deque
n=int(input())
for _ in range(n):
    coins={}
    for i in range(ord('A'),ord('L')+1):
        coins[chr(i)]=0# 0=weizhi, '0'=zhengchang,1=zhongle, -1=qingle
    weigh=deque([])
    for i in range(3):
        line=list(input().split())
        if line[2]=='even':
            weigh.appendleft(line)
        else:weigh.append(line)
    for line in weigh:
        if line[2]=='even':
            for i in line[0]:coins[i]='0'
            for i in line[1]:coins[i]='0'
        elif line[2]=='up':
            for i in line[0]:
                if coins[i]!='0':coins[i]+=1
                if coins[i]==0:coins[i]='0'
            for i in line[1]:
                if coins[i]!='0':coins[i]-=1
                if coins[i]== 0: coins[i] = '0'
        elif line[2]=='down':
            for i in line[1]:
                if coins[i]!='0':coins[i]+=1
                if coins[i]==0:coins[i]='0'
```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```
from collections import deque
n=int(input())
for _ in range(n):
    coins={}
    for i in range(ord('A'),ord('L')+1):
        coins[chr(i)]=0# 0=weizhi, '0'=zhengchang,1=zhongle, -1=qingle
    weigh=deque([])
    for i in range(3):
        line=list(input().split())
        if line[2]=='even':
            weigh.appendleft(line)
        else:weigh.append(line)
    for line in weigh:
        if line[2]=='even':
            for i in line[0]:coins[i]='0'
            for i in line[1]:coins[i]='0'
        elif line[2]=='up':
            for i in line[0]:
                if coins[i]!='0':coins[i]+=1
                if coins[i]==0:coins[i]='0'
            for i in line[1]:
                if coins[i]!='0':coins[i]-=1
                if coins[i]==0: coins[i] = '0'
        elif line[2]=='down':
```

基本信息

#: 47838021
题目: 02692
提交人: 24n2400011782
内存: 3668kB
时间: 22ms
语言: Python3
提交时间: 2024-12-19 15:03:30

01088: 滑雪

dp, dfs similar, <http://cs101.openjudge.cn/practice/01088>

思路:

dfs+dp太慢了...(不过放了lru_cache之后也差不多快了)

参考了算法基础与在线实践的解法:

每个点的最长路线应该是它周围四个点最长的最长路径再+1,但处理顺序是个问题,因为如果从左到右从上到下遍历的话会出现如下情况:

1 2 3 4 5

6 7 8 1 2

8 6 5 4 3

在7那个点的最长路径应该是7654321,但是由于处理到7的时候它底下那个6还没处理,所以会漏掉这个路径.

所以只需按高度从小到大处理就可以了.这可以用heapq实现:

在小顶堆h中存储矩阵中每个点的高度及位置,然后依次弹出来处理

快了10倍...

代码:

```
import heapq

mat=[]
h=[]
r,c=map(int,input().split())
for i in range(r):
    mat.append(list(map(int,input().split())))
    for j in range(c):
        heapq.heappush(h,(mat[i][j],i,j))
dp=[[1]*c for _ in range(r)]
d=[(-1,0),(0,1),(1,0),(0,-1)]
ans=1
while h:
    height,x,y=heapq.heappop(h)
    for dx,dy in d:
        nx,ny=x+dx,y+dy
        if 0<=nx<r and 0<=ny<c and mat[nx][ny]<height and dp[nx][ny]+1>dp[x][y]:
            dp[x][y]=dp[nx][ny]+1
            ans=max(ans,dp[x][y])
print(ans)
```

代码运行截图 == (至少包含有"Accepted") ==

状态: Accepted

源代码

```
import heapq

mat=[]
h=[]
r,c=map(int,input().split())
for i in range(r):
    mat.append(list(map(int,input().split())))
    for j in range(c):
        heapq.heappush(h,(mat[i][j],i,j))
dp=[[1]*c for _ in range(r)]
d=[(-1,0),(0,1),(1,0),(0,-1)]
ans=1
while h:
    height,x,y=heapq.heappop(h)
    for dx,dy in d:
        nx,ny=x+dx,y+dy
        if 0<=nx<r and 0<=ny<c and mat[nx][ny]<height and dp[nx][ny]+1>dp[x][y]:
            dp[nx][ny]=dp[x][y]+1
            ans=max(ans,dp[nx][ny])
print(ans)
```

基本信息

#: 47864931
题目: 01088
提交人: 24n2400011782
内存: 4800kB
时间: 55ms
语言: Python3
提交时间: 2024-12-20 19:01:47

25572: 螃蟹采蘑菇

bfs, dfs, <http://cs101.openjudge.cn/practice/25572/>

思路:
对螃蟹的身体加以考虑就行了

代码:

```
#pylint:skip-file
from collections import deque
def bfs(x1,y1,x2,y2,end):
    global mat
    q=deque([(x1,y1,x2,y2)])
    d=[(0,1),(0,-1),(-1,0),(1,0)]
    visited=set()
    while q:
        x1,y1,x2,y2=q.popleft()
        for dx,dy in d:
            visited.add((x1,y1,x2,y2))
            if end in [(x1,y1),(x2,y2)]:
                return 'yes'
            nx1,ny1,nx2,ny2=x1+dx,y1+dy,x2+dx,y2+dy
            if 0<=nx1<len(mat) and 0<=nx2<len(mat) \
                and 0<=ny1<len(mat[0]) and 0<=ny2<len(mat[0]):
                if (nx1,ny1,nx2,ny2) not in visited \
                    and 1 not in (mat[nx1][ny1],mat[nx2][ny2]):
                    q.append((nx1,ny1,nx2,ny2))
    return 'no'

n=int(input())
mat=[]
start=[]
end=()
for i in range(n):
    mat.append(list(map(int,input().split())))
```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```
#pylint:skip-file
from collections import deque
def bfs(x1,y1,x2,y2,end):
    global mat
    q=deque([(x1,y1,x2,y2)])
    d=[(0,1),(0,-1),(-1,0),(1,0)]
    visited=set()
    while q:
        x1,y1,x2,y2=q.popleft()
        for dx,dy in d:
            visited.add((x1,y1,x2,y2))
            if end in [(x1,y1),(x2,y2)]:
                return 'yes'
            nx1,ny1,nx2,ny2=x1+dx,y1+dy,x2+dx,y2+dy
            if 0<=nx1<len(mat) and 0<=nx2<len(mat) \
                and 0<=ny1<len(mat[0]) and 0<=ny2<len(mat[0]):
                if (nx1,ny1,nx2,ny2) not in visited \
                    and 1 not in (mat[nx1][ny1],mat[nx2][ny2]):
                    q.append((nx1,ny1,nx2,ny2))
    return 'no'

n=int(input())
mat=[]
start=[]
end=[]
for i in range(n):
    mat.append(list(map(int,input().split())))
while 5 in mat[0]:
```

基本信息

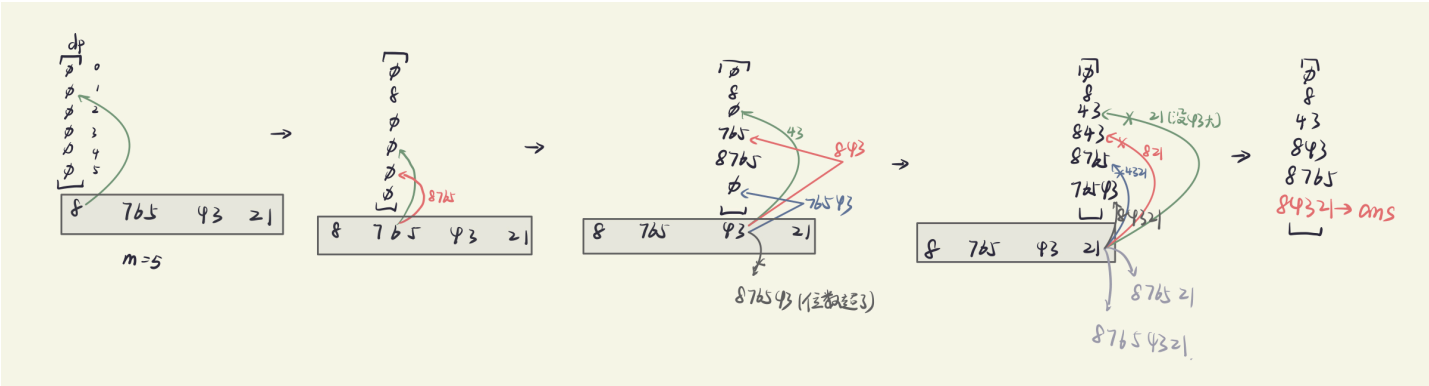
#: 47536490
题目: 25572
提交人: 24n2400011782
内存: 3744kB
时间: 31ms
语言: Python3
提交时间: 2024-12-03 17:15:40

27373: 最大整数

dp, <http://cs101.openjudge.cn/practice/27373/>

思路:

- 1. 先把这些数按"质量"降序排列,其中"质量"高是指让这个数放在前面能获得更大的新数,例如8的质量比84高,因为884>848.这样我们就能在后面的操作中,尽可能让高质量数排在前面.
[记a的质量比b高为big(a,b),a的质量比b低为small(a,b).
可以证明,"质量"的高低具有传递性,即big(a,b) and big(b,c)=>big(a,c).
实质上通过数学方法可以证明,x的"质量"可以用 $x/(10^{*\text{len}(\text{str}(x))-1})$ 表征.]
- 2. 由于本题中数的"位数"是一个需要考虑和控制的量,我们建立一个dp数组,存储"如果组成的数是x位的,那么组成的最大x位数是多少".
- 3. 然后开始遍历:对于输入的数组arr中的每一个数,首先尝试在dp中添加它自己,再分别尝试在dp中添加(dp[i]+str(它自己)).
文字写不清楚...上图



代码:

```
m=int(input())
n=int(input())
arr=list(map(int,input().split()))
arr.sort(key=lambda x:x/(10**len(str(x))-1),reverse=True)
```

```

dp=['']*205
ans=0
for i in arr:
    tmp={len(str(i)):str(i)}
    for j in dp:
        if len(str(j)+str(i))<=m:
            tmp[len(j+str(i))]=str(j)+str(i)
    for k,v in tmp.items():
        dp[k]=str(max(dp[k],str(v)))
        ans=max(ans,int(dp[k]))
print(ans)

```

代码运行截图 （至少包含有"Accepted"）

状态: Accepted

源代码

```

m=int(input())
n=int(input())
arr=list(map(int,input().split()))
arr.sort(key=lambda x:x/(10**len(str(x))-1),reverse=True)
dp=['']*205
ans=0
for i in arr:
    tmp={len(str(i)):str(i)}
    for j in dp:
        if len(str(j)+str(i))<=m:
            tmp[len(j+str(i))]=str(j)+str(i)
    for k,v in tmp.items():
        dp[k]=str(max(dp[k],str(v)))
        ans=max(ans,int(dp[k]))

print(ans)

```

基本信息

#: 47875557
 题目: 27373
 提交人: 24n2400011782
 内存: 3664kB
 时间: 415ms
 语言: Python3
 提交时间: 2024-12-21 12:03:39

02811: 熄灯问题

brute force, <http://cs101.openjudge.cn/practice/02811>

思路： 一直按照题里提示3的方法做下去总能到的.

先从上到下一行行点一遍,再从下到上,再从左到右,再从右到左,然后循环
 至于为什么我也不知道...

p.s.github上那个熄灯游戏这样做貌似会死循环,不知道它生成数据的时候是怎么做的
 p.p.s.顺便附上一段生成测试数据的代码,用这个代码生成的数据都可解
 代码:

```

import sys
input=sys.stdin.readlines()
mat=[]
for line in input:
    mat.append(list(map(int,line.split())))
sol=[[0]*len(mat[0]) for _ in range(len(mat))]
def click(x,y):
    sol[x][y]=int(not sol[x][y])
    mat[x][y]=int(not mat[x][y])
    d=[(1,0),(-1,0),(0,1),(0,-1)]
    for dx,dy in d:
        if 0<=x+dx<len(mat) and 0<=y+dy<len(mat[0]):
            mat[x+dx][y+dy]=int(not mat[x+dx][y+dy])
while True:
    '''
    for _ in sol:
        print(*)<br>
        print('_____')
    '''
    for i in range(0,len(mat)-1):

```

```

        for j in range(0,len(mat[0])):
            if mat[i][j]==1:
                click(i+1,j)

    if 1 not in mat[-1]:break
    for i in range(len(mat)-1,0,-1):
        for j in range(0,len(mat[0])):

#熄灯问题数据生成.生成两个矩阵,第一个是题,第二个是解
import random

mat=[[0]*6 for _ in range(5)]
sol=[[0]*6 for _ in range(5)]
def click(x,y):
    sol[x][y]=int(not sol[x][y])
    mat[x][y]=int(not mat[x][y])
    d=[(1,0),(-1,0),(0,1),(0,-1)]
    for dx,dy in d:
        if 0<=x+dx<len(mat) and 0<=y+dy<len(mat[0]):
            mat[x+dx][y+dy]=int(not mat[x+dx][y+dy])
for i in range(random.randint(10,15)):
    x=random.randint(0,4)
    y=random.randint(0,5)
    click(x,y)
for _ in mat:
    print(*)
print('-----')
for _ in sol:
    print(*)

```

代码运行截图（至少包含有"Accepted"）

状态: Accepted

源代码

```

import sys
input=sys.stdin.readlines()
mat=[]
for line in input:
    mat.append(list(map(int,line.split())))
sol=[ [0]*len(mat[0]) for _ in range(len(mat))]
def click(x,y):
    sol[x][y]=int(not sol[x][y])
    mat[x][y]=int(not mat[x][y])
    d=[(1,0),(-1,0),(0,1),(0,-1)]
    for dx,dy in d:
        if 0<=x+dx<len(mat) and 0<=y+dy<len(mat[0]):
            mat[x+dx][y+dy]=int(not mat[x+dx][y+dy])
while True:
    '''
    for _ in sol:
        print(*)
    print('_____')
    '''
    for i in range(0,len(mat)-1):
        for j in range(0,len(mat[0])):
            if mat[i][j]==1:
                click(i+1,j)

    if 1 not in mat[-1]:break
    for i in range(len(mat)-1,0,-1):

```

基本信息

#: 47874333
 题目: 02811
 提交人: 24n2400011782
 内存: 3776kB
 时间: 24ms
 语言: Python3
 提交时间: 2024-12-21 11:12:31

08210: 河中跳房子

binary search, greedy, <http://cs101.openjudge.cn/practice/08210/>

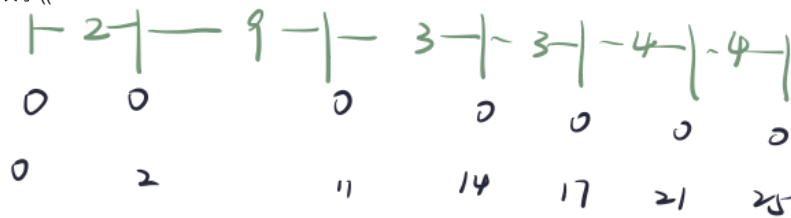
思路:

看了月度开销的题解...

值得一提的是这题,月度开销,Aggressive cows都是同类题,而且主角都是FJ

不禁让人好奇FJ到底有什么故事((

distances:



n 块石头, 相当于 $(n+1)$ 个间距.

移去 m 块, 剩下 $(n+m-1)$ 个间距.

假设剩下的间距中最短的为 x .

则希望 x 尽量大.

- 我们事先预期一个 x 值, 看看这个 x 值能否实现.

如果实现, 就得寸进尺, 把 x 调大;

反之, 则调小.

(就像跳高运动, 先预设一个横杆的高度,

跳过了就调高, 没过就调低).

- 对于调整 x 的方式, 采用高效的二分.

↳ $\min(\text{distances})$ 为 low , $\sum(\text{distances})$ 为 hi .

- 对于检查该 x 能否实现:

对距离进行分段, 连续的几个距离为一段.

每段距离应恰好 $\geq x$. 若分出了 $\geq n+m-1$ 段, 则

移去的石头不多于 m 个, 即能实现.

代码:

```
l, n, m = map(int, input().split())  
r = [0]
```

```
distances=[]
for i in range(n):
    r.append(int(input()))
r.append(l)
for i in range(1,n+2):
    distances.append(r[i]-r[i-1])
#检查FJ期望"最小距离不低于x"能否实现
def check(x):
    num,cut=0,0
    for i in range(n+1):
        if cut+distances[i]>=x:
            num+=1
            cut=0
        else:cut+=distances[i]
    if num>=n-m+1:
        return True
    else:return False
#通过二分(啊不是,二分)查找调整FJ的期望
low,high=min(distances),sum(distances)
while high-low >0:
    mid = (low + high) // 2
    #print(low,high)
    #print('mid',mid)
    if check(mid):
        low=mid
    else:high=mid
```

代码运行截图（至少包含有"Accepted"）

状态: Accepted

源代码

```
l,n,m=map(int,input().split())
r=[0]
distances=[]
for i in range(n):
    r.append(int(input()))
r.append(l)
for i in range(1,n+2):
    distances.append(r[i]-r[i-1])
#检查FJ期望"最小距离不低于x"能否实现
def check(x):
    num,cut=0,0
    for i in range(n+1):
        if cut+distances[i]>=x:
            num+=1
            cut=0
        else:cut+=distances[i]
    if num>=n-m+1:
        return True
    else:return False
#通过二分查找调整FJ的期望
low,high=min(distances),sum(distances)
while high-low >0:
    mid = (low + high) // 2
    #print(low,high)
    #print('mid',mid)
```

基本信息

#: 47870661
题目: 08210
提交人: 24n2400011782
内存: 8028kB
时间: 281ms
语言: Python3
提交时间: 2024-12-21 00:44:55

2. 学习总结和收获

如果作业题目简单，有否额外练习题目，比如：OJ“计概2024fall每日选做”、CF、LeetCode、洛谷等网站题目。

Cheatsheet做好了。

感觉能想出来题目完全得看当天的精神状态,而精神状态似乎是个随缘的东西...最大整数那题真的一晚上想不出来第二天灵光一现就出来了
转眼是最后一次作业了(除了AssignmentP).写了一篇总结,送给CS101以及这学期学过的其他课.

Cheatsheet及总结在附件中。

CS是门很好玩的学科,各种意义上的好玩,也很大程度重构了我的思维方式((

完结撒花!